



# TOIMINNANOHJAUSJÄRJESTEL- MÄN KÄYTÖN ANALYSOINTI JA SIITÄ JOHDETTAVA BI ASIAKAS- YRITYKSEN JOHDOLLE

Koulutusala Tekniikan ja liikenteen ala			
Koulutusohjelma/Tutkinto-ohjelma Tietotekniikan tutkinto-ohjelma			
Työn tekijä(t) Tatu Horto			
Työn nimi			
Toiminnanohjausjärjestelmän käytön analysointi ja siitä johdettava BI asiakasyritysten johdolle.			
Päiväys	17.5.2018	Sivumäärä/Liitteet	31/0
Ohjaaja(t)			
Jukka Kinnunen, Keijo Kuosmanen			
Toimeksiantaja/Yhteistyökumppani(t)			
Mastercom Oy			
<p>Tiivistelmä</p> <p>Opinnäytetyön tavoitteena oli tutkia toimeksiantajan toiminnanohjausjärjestelmän käyttöä ja kehittää tutkimuksen pohjalta yhteenvetosovellus toimeksiantajan asiakkaiden käyttöön. Yhteenvetosovelluksen on tarkoitus toimia Business Intelligence -työkaluna asiakasyrityksille. Opinnäytetyö jaettiin kahteen osioon, tutkimus- ja toteutusosaan. Päätettiin, että toteutusosan rajausta tehdään opinnäytetyön edetessä, riippuen perehdytyksen sekä tutkimusosan laajuudesta ja itse toteutuksen etenemisestä.</p> <p>Tutkimusosa aloitettiin perehtymällä toimeksiantajan toiminnanohjausjärjestelmään. Perehtymisvaiheessa otettiin huomioon asiakkaan näkökulma toiminnanohjausjärjestelmän käyttöön ja näin pystyttiin arvioimaan, mikä tieto toiminnanohjausjärjestelmässä ja sen käytössä on keskeistä. Perehdytysvaiheen jälkeen siirryttiin konkreettisesti tutkimaan ja listaamaan tietoja, jotka ovat toiminnanohjausjärjestelmän käytön kannalta keskeisiä. Tutkittiin myös, missä kyseinen tärkeä tieto toiminnanohjausjärjestelmän taustalla olevassa tietokannassa sijaitsee ja miten se on sieltä hyödynnettävissä.</p> <p>Tutkimusosan yksi keskeisistä työvaiheista oli myös aiheeseen liittyvään teoriaan perehtyminen. Business Intelligence ja toiminnanohjausjärjestelmät ovat keskeisiä asioita nykypäivän liiketoiminnassa, joten teoriaosassa tutustuttiin erityisesti niihin liittyvään kirjallisuuteen ja julkaisuihin. Lisäksi perehdyttiin tietokantojen teoriaan, sillä sekä Business Intelligencen että toiminnanohjausjärjestelmien pohjana ovat erilaiset tietokantaratkaisut. Opinnäytetyön raportista tehtiin vahvasti teoriapohjainen.</p> <p>Toteutusosassa tehtiin SQL-kyselyitä tietokantaan tutkimusvaiheessa selvitettyjen ja listattujen tietojen pohjalta. SQL-kyselyt tehtiin niin, että jokaista tärkeäksi listattua tietoa vastasi oma kyselynsä, joka palauttaa kyseisen tärkeän tiedon tietokannasta. SQL-kyselyiden pohjalta kehitettiin PHP-moduulit, jotka toimivat lopullisen yhteenvetojärjestelmän sekä SQL-kyselyiden välikappaleina. Toteutusosa rajattiin PHP-moduuleihin.</p> <p>Opinnäytetyössä saatiin tehtyä pohja yhteenvetosovellukselle. Yhteenvetosovellusta kehitetään opinnäytetyön ulkopuolella ja tarkoituksena on, että se pystyy hyödyntämään opinnäytetyön toteutuksessa kehitettyjä PHP-moduuleja mahdollisimman saumattomasti. Lopullisen yhteenvetosovelluksen on tarkoitus tukea toimeksiantajan yrityksen liiketoimintaa parantamalla sen asemaa markkinoilla sekä vankistamalla jo olemassa olevia asiakassuhteita.</p>			
<p>Avainsanat</p> <p>Toiminnanohjausjärjestelmä, Business Intelligence, Tietokanta</p>			

Field of Study Technology, Communication and Transport			
Degree Programme Degree Programme in Information Technology			
Author(s) Tatu Horto			
Title of Thesis Analysis of an ERP system and the derived BI from it for the customer companies' management			
Date	23 may 2018	Pages/Appendices	31/0
Supervisor(s) Mr. Jukka Kinnunen, Senior Lecturer and Mr. Keijo Kuosmanen, Senior lecturer			
Client Organisation /Partners Mastercom Oy			
<p>Abstract</p> <p>The goal of this thesis was to study and analyse the commissioner's Enterprise Resource Planning system, ERP, and develop a summary program for their customers based on the research. The summary program is supposed to be used as a Business Intelligence tool, BI, by the customers. The thesis was divided into research and implementation parts. The extent of the implementation part would be defined depending on how large the research will be and how the implementation progresses.</p> <p>The research part was started by an introduction to the commissioner's enterprise resource planning system. During the introduction, the customer's point of view was taken into account when studying the use of the ERP system which helped to determine which data in the ERP system and in its use is important. After introduction the actual research was started to determine the important data and during the research the data was listed. Also, the location of the data in database of ERP system was researched.</p> <p>One of the main phases of the research part was to study the theory related to the subject of the thesis. BI and ERP systems are essential parts of businesses today, so the theory part focused on literature and releases related to them. Also, the theory of databases was a big part of studying the theory as databases are a fundamental base to both BI and ERP systems. The report of the thesis is mostly theory based.</p> <p>The actual SQL queries based on the research were made in the implementation part of the thesis. The SQL queries were made so that each data listed as important had its own query which returned the data needed. After that the PHP modules were developed based on the SQL queries. The PHP modules serve as the links between the final program and the SQL queries. The implementation part was limited to cover the PHP modules.</p> <p>The outcome of the thesis was a base for the summary program. The summary program will be developed separately after the completion of this thesis and the goal is that it will use the PHP modules which were developed in the implementation part of the thesis. The final summary program is supposed to support commissioner's business by improving their position in the industry and by strengthening their customer relationships.</p>			
<p>Keywords</p> <p>Enterprise resource planning system, Business Intelligence, Database</p>			

## ESIPUHE

Tahdon kiittää Kihon toimitusjohtaja Antti Koposta hänen tarjoamistaan opinnäytetyöaiheista. Kävimme palaverin, jossa päätimme kahden aiheen väliltä ja teimme linjauksia, mitä opinnäytetyö pitää sisällään. Kiitän lisäksi Ville Nuutista, joka jo valintavaiheessa auttoi rajaamaan ja määrittelemään aihetta sekä perehdytti minua muiden ohella yrityksen työympäristöön.

Suurkiitos myös kaikille ystävilleni ja läheisilleni, jotka ovat tukeneet minua opinnäytetyötä tehdessäni sekä muutoin auttaneet saattamaan opintoni valmiiksi.

Kuopiossa 8.5.2018

Tatu Horto

## SISÄLTÖ

SISÄLTÖ .....	5
1 JOHDANTO .....	7
2 MASTERCOM OY .....	8
2.1 Kiho-tuoteperhe.....	8
2.2 Järjestelmän sovellukset.....	9
3 TIETOKANTA .....	11
3.1 Relaatiomalli.....	11
3.1.1 Taulut .....	11
3.1.2 Avaimet.....	12
3.2 Tietokantojen käyttötapoja .....	12
3.3 Normalisointi .....	13
3.3.1 Normaalimuodot .....	13
3.4 Structured Query Language.....	14
3.4.1 SQL:n käyttötavat .....	14
3.4.2 SQL-kyselyt .....	15
4 BUSINESS INTELLIGENCE .....	17
4.1 BI-ratkaisut yleisesti.....	17
4.2 BI-ratkaisujen tekniikka .....	18
4.2.1 On-Line Analytical Processing.....	19
4.2.2 FASMI-määritelmä.....	20
4.2.3 Tietovarastot .....	20
4.3 Business Intelligencen merkitys .....	21
5 TOIMINNANOHJAUSJÄRJESTELMÄ.....	23
5.1 Toiminnanohjausjärjestelmien merkitys ja keskeiset ominaisuudet .....	23
6 TUTKIMUS JA TOTEUTUS.....	24
6.1 Perehdytys ja tutkimus.....	24
6.2 Tietokanta .....	25
6.3 Sovelluskehitys .....	26
6.3.1 Dokumentointi .....	27
6.3.2 Testaus .....	28
7 YHTEENVETO.....	29

7.1	Jatkokehitys .....	29
7.2	Henkilökohtaisen osaamisen kasvu.....	29
7.3	Tavoitteiden saavuttaminen.....	30
LÄHTEET JA TUOTETUT AINEISTOT .....		31

## 1 JOHDANTO

Nykypäivän yritysmaailmassa on keskeistä, että yrityksen toiminnan kannalta tärkeää tietoa analysoidaan ja jalostetaan hyödynnettävään muotoon. Jalostetusta tiedosta voidaan kehittää erilaisia yhteenvetoja, joiden avulla pystytään optimoimaan toimintaketjuja ja tekemään tarkempia suunnitelmia siitä, mihin suuntaan yritystä kehitetään ja mihin toiminnalla tähdätään. Toiminnanohjausjärjestelmät ovat yleistymässä yrityksissä ja niiden avulla saadaan merkittävää hyötyä yrityksen tehokkuuteen ja tuottavuuteen. Business Intelligencessä on kyse tiedonkeruusta, minkä avulla pystytään luomaan ennusteita tulevista markkinoista. (Logistiikan maailma, Kauppapuisto, Itewiki.)

Opinnäytetyö koostuu kahdesta osasta: Tutkimus- ja toteutusosasta. Tutkimusosan on tarkoitus toimia pohjana toteutusosalle. Toteutusosassa on tarkoituksena kehittää yhteenvetojärjestelmä tutkimusosassa selvinneiden tietojen ja teorian pohjalta. Opinnäytetyön laajuus ja rajausta määräytyvät työn edetessä riippuen siitä, kuinka paljon työtä kukin työvaihe vaatii. Raportissa käytetään myös termiä yhteenvetosovellus, mutta silloin viitataan kuitenkin pelkästään ohjelmoitavaan ohjelmaan, kun taas järjestelmällä tarkoitetaan yhteenvetojen toiminnallisuutta kokonaisuudessaan.

Opinnäytetyön tutkimusosassa selvitetään, mikä tieto toimeksiantajan kehittämän toiminnanohjausjärjestelmän tietokannoissa on tärkeää asiakasyritysten kannalta ja kuinka tuota tietoa voidaan hyödyntää. Tutkimuksen pohjalta on tarkoitus kehittää järjestelmä, jolla voidaan tuottaa asiakasyritysten johdolle yhteenvetoja hankkimansa toiminnanohjausjärjestelmän käytöstä. Tutkimusosan keskeistä vaiheista on myös aiheeseen liittyvän teoriaan perehtyminen, jotta yhteenvetojärjestelmä voitaisiin suunnitella ja toteuttaa mahdollisimman kestävästi ja kattavasti.

Opinnäytetyön toteutusosassa kehitetään SQL-kyselyitä, joiden avulla saadaan haettua tutkimusosassa selvinneitä tärkeitä tietoja tietokannasta. SQL-kyselyiden lisäksi toteutuksessa on tarkoituksena kehittää PHP-moduulit, jotka toimivat välikappaleina lopullisen yhteenvetosovelluksen ja tietokannan välillä. Toteutukseen kuuluu myös mahdollisesti varsinaisen yhteenvetosovelluksen kehitys.

Työn tavoitteena on tuottaa toimeksiantajalle tarpeenmukainen tutkimus pohjatyönä heidän visioimalleen järjestelmälle. Opinnäytetyön on tarkoitus tulla valmiiksi kevään 2018 aikana. Järjestelmälle ei ole määriteltyä tarkempaa käyttöönottoaikaa, mutta pohjatyönä tehtävän tutkimuksen tulee olla valmis ennen kesää. Raportissa käydään läpi tutkimuksen eri työvaiheet, niissä ilmenneet asiat ja tutkimusongelmat sekä -tulokset.

## 2 MASTERCOM OY

Mastercom Oy, aputoiminimeltään Kiho, tuottaa automatisoituja tiedonkeräyspalveluita asiakasyrityksilleen ja on nykyään yksi alan johtavista yrityksistä Suomessa. Yrityksen tarjoamien palveluiden tavoitteena on vähentää paperityötä ja tehostaa tiedonkulkua asiakasyrityksissä. Vuonna 2003 Pohjois-Savossa perustettu yritys aloitti toimintansa ICT-tukipalvelutuottajan sekä ohjelmistoalihankkijana. Pari vuotta myöhemmin alettiin kehittämään GPS-Msolutions nimistä ajoneuvopaikannuspalvelua ja yritys lopetti ICT-tukipalveluiden tarjoamisen vuonna 2007, jolloin toiminnan keskipiste siirrettiin ajoneuvopaikannuspalveluun. Mastercom Oy tunnetaan nykyään aputoiminimellä Kiho. (Kiho 2018a.)

### 2.1 Kiho-tuoteperhe

Mastercom Oy otti Kihon aputoiminimekseen vuonna 2010. Kiho tarkoittaa myös Kiho-tuoteperhettä, joka on yrityksen nykyään tarjoama palvelu. Kiho-tuoteperhe on usean eri toiminnallisuuden kokonaisuus, joiden avulla asiakasyritys voi hallita liiketoiminnan eri osa-alueita, joten sitä voidaan kutsua toiminnanohjausjärjestelmäksi. (Kiho 2018a, Kiho 2018b.)

Asiakasyritys voi palvelun hankkiessaan valita, mitä ominaisuuksia palveluun haluavat. Hinnat vaihtelevat eri tuotepakettien välillä. Ajopäiväkirja on Kihon tarjoamista palveluista edullisin, joka mahdollistaa tietojen keräämisen ajoneuvojen ajoista automaattisesti. Ajopäiväkirjan lisäksi asiakkaat voivat hankkia telematiikka -palvelun, jonka avulla he voivat esimerkiksi tarkastella ajoneuvojen sijainteja kartalta, saada käyttöönsä erilaisia raportteja ajoneuvojen käytöstä ja käyttää hälytyksiä ilmoittamaan mahdollisista tapahtumista ajoneuvojen toiminnassa. Muita mahdollisia paketteja tarjotuista palveluista ovat työaika, tehtävät, laskutus, ilmoitustaulu ja pätevyudet. (Kuva 1.)

Vaikka Kihossa ominaisuuksia on liiketoiminnan eri osa-alueilta, järjestelmän pääpaino on ajoneuvojen paikantamisessa. Paikannusjärjestelmä on pohjana ajoneuvojen telematiikkatiedoille. Kerättyä paikkatietoa jalostetaan raportoitavaan muotoon, jonka avulla asiakkaat voivat hyödyntää ajoneuvoistaan saatua tietoa liiketoiminnan tarpeisiin.



KUVA 1. Kiho tuotteiden ominaisuudet sekä hinnat (Kiho 2018c.)

	Ajopäiväkirja 10€ kk/ ajoneuvo	Telematiikka 25€ kk/ajoneuvo	Työaika 15€ kk/työntekijä	Tehtävät 15€ kk/työntekijä	Laskutus 15€ kk/työntekijä	Ilmoitustaulu 20€ kk/näyttö	Pätevyudet 5€ kk /käyttäjä
Live-kartta		x					
Mobiili Kiho		x	x	x			
Reittihistoria		x					
Ajopäiväkirja	x	x					
Ajotapa raportti		x					
Vertailuraportit		x					
Alueraportti		x					
Asiakkaat, kustannuspaikat ja tuotteet		x	x	x	x		
Häilytykset		x					
Tankkaukset		x					
Kalustoluettelo		x					
Rengaspainevalvonta		+7e/kk					
Työaika			x		x		
Työajan siirto rajapinnan kautta			x				
Mobiili Työaika			x				
Työajan raportit			x				
Työajan laskutus			x*		x		
Tehtävät				x			
Tehtävien laskutuksen siirto rajapinnan kautta				x*	x		
Tehtävien raportointi				x			
Tehtävien suorittaminen mobiilissa				x			
Tehtävien laskutus				x*	x		
Myynti- ja ostolaskut					x		
Pätevyudet							x
Ilmoitustaulu/ Mainosnäytöt						x	

\*vaatii myös laskutuksen

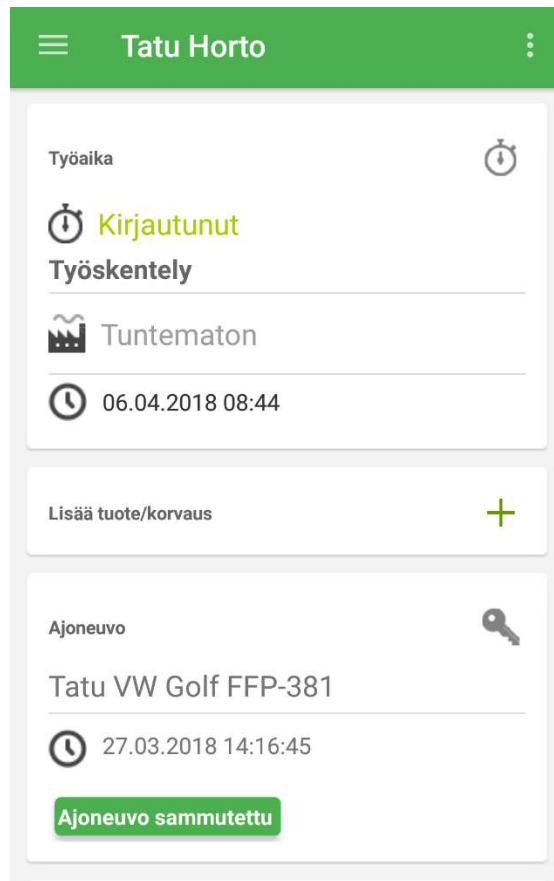
## 2.2 Järjestelmän sovellukset

Kihon kehittämää toiminnanohjausjärjestelmää käytetään selaimessa, sillä se on alun perin web-sovellus. Tämä tarkoittaa sitä, että käyttäjä tarvitsee tietokoneen tai mobiililaitteen, toimivan internetiyhteyden sekä verkkoselaimen käyttääkseen Kiho-järjestelmää. Web-käyttöliittymässä päästään käsiksi tuotteen kaikkiin ominaisuuksiin, jotka asiakas on valinnut tuotetta hankkiessa, lukuun ottamatta Mobiili Kihon ominaisuuksia. (Kuva 2.)

KUVA 2. Kihojärjestelmän websovelluksen käyttöliittymä.

Kiho-järjestelmästä on kehitetty myös mobiiliversio, joka toimii Android-käyttöjärjestelmillä ja näin mahdollistaa järjestelmän käytön mobiililaitteella. Kyseisellä versiolla käyttäjä pystyy kirjautumaan sisään töihin ja ajoneuvoon. Mobiililaitetta, jossa on Kihon mobiiliversio, voidaan käyttää ajoneuvona eli laitteena, jota paikannetaan. Käyttäjä pystyy myös tarkastelemaan karttaa, luomaan tehtäviä ja hallinnoimaan työaika. (Kuva 3.)

KUVA 3. Kihojärjestelmän mobiilisovelluksen käyttöliittymä.



### 3 TIETOKANTA

Tietokannalla tarkoitetaan tiettyyn käyttötarkoitukseen laadittua joukkoa erilaisista tiedoista, jotka ovat loogisesti yhteenkuuluvia. Tiedot ovat tallennettuja ja niiden on tarkoitus olla käsiteltävissä helposti erilaisilla tietokantakielillä. Tietokannan hallintajärjestelmä on ohjelmisto, jolla hallinnoidaan tietokannassa olevaa tietoa. Tietokantojen hallintajärjestelmät mahdollistavat tietokannan käyttäjille useita palveluita, joiden avulla saadaan lisättyä muutosjoustavuutta, turvattua tietoeheyttä, parannettua suorituskkyä ja yleisesti helpotetaan sovellusohjelmointia. Hallintajärjestelmän ansiosta tietokanta pystyy olemaan yhteiskäyttöinen, ajantasainen, ei-toisteinen ja eheä. (Hovi, Huotari, Lahdenmäki 2005, 4 – 5.)

Relaatiotietokannat, verkkomalliset, hierarkkiset ja oliotietokannat ovat erilaisia tietokantatyyppejä. Nykyään yleisimmin käytössä ovat relaatiotietokannat, jotka ovat käytettävyydeltään helpompia kuin vanhemmat verkkomalliset ja hierarkkiset tietokannat. Tietovarastojen ja erilaisten operatiivisten sovellusten toteutuksessa käytetään relaatiokantoja. Oliotietokannat ovat harvemmin käytettyjä tietokantatyyppejä, jotka eivät vastaa relaatiomallin mukaista rakennetta tietokannoissa. (Hovi ym. 2005, 5-6.)

#### 3.1 Relaatiomalli

Kiho käyttää palvelunsa taustalla relaatiomallista MariaDB -tietokantajärjestelmää. Relaatiomallinen tietokanta pohjautuu tutkija E. F. Coddin v. 1970 määrittelemään relaatiotietokannan teoreettiseen pohjaan, joka perustuu matematiikkaan, joukko-oppiin sekä predikaattilogiikkaan. Perinteiset hierarkkiset ja verkkomalliset tietokantatyypit ovat siirtyneet syrjään relaatiomalliin perustuvien relaatiotietokantatuotteiden tieltä. Relaatiokannoissa käytettävä SQL-tietokantakieli on nykyään vakioitunut lähes ainoaksi tietokantojen käsittelykieleksi. (Hovi ym. 2005, 7-8.)

##### 3.1.1 Taulut

Relaatiomalliin perustuvan tietokannan rakenne koostuu tauluista, jotka ovat tietokantojen peruselementtejä (Hovi ym. 2005, 8 – 9). Sarakkeet eli kentät ovat taulujen osia, jotka kuvaavat erilaisia tallennettavia tietoja. Rivi eli tietue muodostuu sarakkeista ja on näin yksilöllinen tietue taulussa. Taulussa voi olla useita tietueita tai ei yhtään, jolloin kyseessä on tyhjä taulu. (Ekonoja, Lahtonen, Mäntylä 2004.)

Sarakkeet ovat kuin rivien ominaisuuksia. Esimerkkinä ”työntekijät” -taulussa sarakkeina voivat olla ”tunniste”, ”etunimi” ja ”sukunimi”, jotka kaikki ovat yhden työntekijän ominaisuuksia. Tietue puolestaan on taulun yksi instanssi eli esimerkiksi ”työntekijät” -taulun yksi tietue on yksittäinen työntekijä, jonka eri ominaisuudet on määritelty sarakkeissa ja ominaisuuksien arvot kussakin sarakkeessa kyseisin rivin kohdalla.

### 3.1.2 Avaimet

Tauluissa tunnisteena toimii uniikki eli yksilöivä perusavain. Yksilöivyyden vuoksi kahdella rivillä ei perusavaimen sarakkeessa voi olla samaa arvoa. Perusavaimet ja niiden suunnittelu on keskeisessä roolissa koko tietokannan suunnitteluprosessissa. Perusavain ei välttämättä koostu vain yhdestä sarakkeesta, vaan se voidaan määrittää myös useamman sarakkeen avulla. (Hovi ym. 2005, 9.) Mikäli perusavain on muodostettu useammasta kuin yhdestä sarakkeesta, kyseessä on yhdistetty avain. Perusavaimen eheys tarkoittaa sitä, että sillä täytyy olla aina jokin arvo eli se ei voi olla tyhjä. (Ekonoja ym. 2004.)

Tauluilla voi olla myös viiteavain, joka viittaa jonkin toisen taulun perusavaimen. Viiteavainta tarvitaan, kun kahden taulun välillä on riippuvuus. Viiteavainten avulla voidaan tehdä liitoksia eli yhdistää kaksi taulua toisiinsa. (Hovi ym. 2005.) Viite-eheys tarkoittaa yhteyttä, jossa toinen taulu viittaa toisen taulun perusavaimen. Viite-eheyden mukaan jokaisen viittaavan taulun viiteavaimen arvolle pitää löytyä vastaava arvo viitatus taulun perusavaimista. (Ekonoja ym. 2004.)

Perusavain toimii uniikkina tunnisteena taulun riveille. Esimerkiksi ”yritykset” – taulun yhden tietueen eli yrityksen perusavaimena voi olla yritystunnus, eikä toisella yrityksellä voi olla samaa yritystunnusta (kuva 4). Viiteavaimen avulla voidaan yhdistää kaksi taulua toisiinsa. Esimerkiksi ”työntekijät” – taulun yksi tietue eli työntekijä liittyy johonkin yritykseen, joten sen viiteavaimena on tietyn ”yritykset” – taulun yrityksen perusavain (kuva 4). Yhdellä työntekijällä voi olla vain yksi yritys, mutta yrityksellä voi olla useita työntekijöitä. Tämän tyyppistä yhteyttä kahden taulun välillä kutsutaan yksi–moneen-yhteydeksi. (Hovi ym. 2005, 9.)

KUVA 4. Esimerkkitaulut.

Työntekijät			
hTunnus	etunimi	sukunimi	yTunnus
121212	Mauri	Meikäläinen	123
111111	Maija	Mattilainen	123
170707	Testi	Henkilöinen	500
290990	Erkki	Esimerkki	600
Yritykset			
yTunnus	nimi	sijainti	pVuosi
123	Testi Oy	Kuopio	2000
500	Oppari Ky	Helsinki	2005
600	Meikäläinen tmi.	Kajaani	2010
1000	Esimerkki Oy	Rovaniemi	2015

## 3.2 Tietokantojen käyttötapoja

On hyvin monia erilaisia tapoja käyttää tietokantoja. Esimerkiksi pankin tilitietokantaan kohdistuu paljon tapahtumia lyhyessä ajassa. Siihen saatetaan kirjata tilitapahtumia sekunnissa jopa sata. Kun

kuormitus on näin suurta, vaatii se paljon suorituskkyä tietokantaa käsittelevältä laitteistolta ja hallintajärjestelmältä. (Hovi ym. 2005, 15.)

Pankin tilitietokanta toimii yleensä reaaliajassa. Kun pankkiautomaatilta nostetaan rahaa toisella puolella Suomea, tulee sen näkyä heti myös tilille tehdyssä kyselyssä toisessa paikassa. Kuormittavissa tapahtumankäsittelyissä käytetään usein myös eräajoja. Eräajoissa kyse on siitä, ettei ihminen vaikuta sen toimintaan eli se tapahtuu automaattisesti. Esimerkiksi tiliotteen tulostus on tapahtuma, joka tehdään eräajona. (Hovi ym. 2005, 15.)

Kun tietokannasta käsitellään kerralla vain tiettyä pienempää osaa, puhutaan operatiivisesta tietojenkäsittelystä. Vain tietyn osan käsittely kerrallaan on tärkeää, kun käsitellään suuria ja kuormittavia tietokantoja. Operatiivisen tietojenkäsittelyn keskeisiä piirteitä ovat myös tietojen tiheä päivittäminen ja niiden pitäminen reaaliaikaisena. (Hovi ym. 2005, 15.)

### 3.3 Normalisointi

Normalisointi tarkoittaa tietokannassa olevien tietojen järjestämistä ja organisointia. Normalisointi on prosessi, jolla vähennetään tietojen redundanssia eli päällekkäisyyttä ja epäyhtenäisiä riippuvuussuhteita. Jotta tietokantaa voitaisiin käyttää mahdollisimman tehokkaasti, se tulee normalisoida. Normalisoinnin tasossa on eri asteita eli muotoja, jotka kertovat, kuinka monta normalisoinnin sääntöä kyseinen tietokanta täyttää. Kolmannetta normaalimuotoa pidetään riittävänä normalisoinnin tasona suurimalle osalle järjestelmistä. Normalisointiprosessissa joudutaan usein tekemään lisää tauluja, joten osa sovelluksenkehittäjistä kokee normalisoinnin työlääksi ja näin ollen normalisoinnin toteutus voi jäädä vajaaksi. (Microsoft 2017.)

#### 3.3.1 Normaalimuodot

Normalisoinnin eri tasot on nimetty normaalimuodon mukaan ensimmäinen normaalimuoto, toinen normaalimuoto ja niin edelleen. Ensimmäinen normaalimuoto tarkoittaa, että yksittäisistä tauluista on poistettu toistuvat ryhmät. Lisäksi se määrittää, että liittyvien tietojen joukolle on luotu oma taulunsa, ja että kyseiset joukot on merkitty perusavaimen avulla. Toisen normaalimuodon mukaan arvojen joukoille, jotka koskevat useita tietueita, tulee olla erilliset taulut. Se määrittelee myös, että näiden taulujen väliset suhteet muodostetaan viiteavaimen avulla. Kolmannessa normaalimuodossa avaimesta riippumattomat kentät on poistettu. (Microsoft 2017.)

Kolmatta normaalimuotoa ei aina ole käytännöllistä noudattaa, joten sitä tulee soveltaa tilanteen mukaan. Yleisesti ottaen kolmas normaalimuoto toimii parhaiten, kun käytettävät tiedot muuttuvat paljon. Kolmen normaalimuodon lisäksi on myös olemassa neljäs ja viides normaalimuoto. On mahdollista, etteivät neljännen ja viidennen normaalimuodon mukaan suunnitellut tietorakenteen tuota täydellistä lopputulosta, joten niitä käytetään harvemmin. (Microsoft 2017.)

Ensimmäinen normaalimuoto tarkoittaa, ettei tietokannan taulussa esiinny toistuvia ryhmiä. "Työntekijät" -taulussa voi olla esimerkiksi "työnantaja 1" ja "työnantaja 2" -kentät, mikä ei täytä ensimmäisen normaalimuodon ehtoa. Jotta ensimmäisen normaalimuodon ehto saadaan täyttymään, tulee työnantajille tehdä oma taulunsa, josta viitataan "työntekijät" -tauluun tai toisinpäin.

Toisen normaalimuodon mukaan tietueen ei tule olla riippuvainen muusta kuin taulun perusavaimesta. "Työntekijät" -taulussa voi olla kenttä työntekijän osoitteesta, mutta samaa osoitetta tarvitaan myös mahdollisesti muissa tauluissa. Toisen normaalimuodon mukaan tulee siis sijoittaa osoitteet omaan tauluunsa ja viitata "työntekijät" -taulusta tietyn työntekijän kohdalla tiettyyn osoitteeseen "Osoitteet" -taulussa. Toisaalta, muut osoitetta tarvitsevat taulut voivat viitata "työntekijät" -taulussa sijaitsevaan tietoon osoitteesta, joten erillistä "Osoitteet" -taulua ei välttämättä tarvita. Keskeistä kuitenkin on, että itse osoite on tallennettuna vain yhteen paikkaan ja muut sitä tarvitsevat tietueet viittaavat siihen.

Kolmannen normaalimuodon mukaan avaimesta riippumattomat kentät tulee poistaa. Esimerkiksi "työntekijät" -taulussa voi olla tietoja työntekijän harrastuksesta. Tiedot työntekijän harrastuksesta eivät kuitenkaan liity itse työntekijän tietoihin, joten työntekijöiden harrastuksille on syytä luoda oma taulunsa, johon "työntekijät" -taulusta viitataan.

### 3.4 Structured Query Language

SQL eli Structured Query Language on IBM:n 70-luvulla kehittämä tietokantakieli, joka on nykyään käytössä kaikissa merkittävässä relaatiotietokantatuotteissa. SQL:n avulla voidaan käsitellä tietokannan rivejä poistamalla tai lisäämällä niitä ja se mahdollistaa myös tietokannan lukemisen ja päivittämisen. SQL on ei-proseduraalinen eli se ei kuvaa sitä, miten tietoa haetaan vaan sitä, mitä tietoa haetaan. (Hovi ym. 2005, 10.)

#### 3.4.1 SQL:n käytötavat

Nimestään huolimatta SQL ei ole vain kyselykieli, sillä sen avulla voidaan käsitellä tietokantaa eri tavoin. Rivejä palauttavien kyselyiden lisäksi SQL:ää voidaan käyttää tietokannan tietojen päivityksiin, muutoksiin ja poistoihin sekä sen rakenteen määrittelyyn ja muuttamiseen. Sillä voidaan myös ohjata tapahtumakäsittelyä ja hoitaa valtuuksia sekä turvallisuutta. SQL tukee API-rajapintoja eri ohjelmointikieliin sekä upotettua SQL:ää ja kohdistimien hallintaa. (Hovi 2009, 14.)

DML eli Data Manipulation Language on SQL:n osa, jolla voidaan tehdä kyselyitä, lisäyksiä, päivityksiä ja poistoja. DDL eli Data Definition Language on puolestaan osa, jonka avulla voidaan määritellä ja muuttaa tietokannan rakennetta. Erilaiset käyttöliittymät mahdollistavat SQL-kyselyiden tekemisen vuorovaikutteisesti, jolloin kyselyn vastaukset saadaan erilliseen ikkunaan. Vuorovaikutteista kyselyiden tekemistä vaaditaan erityisesti silloin, kun kyselyt ovat satunnaisia ja ennalta arvaamattomia. (Hovi 2009, 14.)

SQL:ää voidaan käyttää myös ohjelmoinnin osana upottamalla sitä ohjelmointikieleen, jolloin kyse on upotetusta SQL:stä. SQL-käskyjä voidaan upottaa koodiin sellaisenaan tai käyttää erilaisia rajapintoja, jotka mahdollistavat ohjelman linkittämisen tietokantaan. SQL:n upottaminen ohjelmointikielen osaksi tarkoittaa yksinkertaisesti sitä, että tietokannasta haetut vastaukset saadaan ohjelmointikielen muuttujiin. (Hovi 2009, 16.) Opinnäytetyön toteutuksessa käytettiin upotettua SQL:ää osana PHP-moduuleja.

Kolmitasoarkkitehtuuri on nykyisin yleistynyt tietojärjestelmäarkkitehtuuri. Tämän arkkitehtuurin mukaan tietyn järjestelmän osat toimivat eri palvelimilla. Esimerkkinä on sovellus, jota käytetään asiakkaan työasemalla, mutta sen taustatoiminnot eli itse sovellus on erillisellä palvelimella ja sen käyttämä tietokanta omalla tietokantapalvelimellaan. SQL:n käyttö on erityisen tärkeää asiakas-palvelin-järjestelmissä. Kyseisissä järjestelmissä asiakkaiden työasemilla olevilta ohjelmilta lähetetään SQL-lauseita verkon välityksellä erilliselle palvelimelle, jossa tietokanta on. Kysely lähettää asiakkaan työaseman ohjelmalle esimerkiksi SQL-kyselystä palautuneet rivit. (Hovi 2009, 16.)

SQL on tärkeä osa tietokannan tiedoista tuotettavia raportteja. Vaikka itsessään SQL ei tuota raportteja, sen avulla saatuja hakutuloksia käytetään raporttien muodostamiseen erilaisten työkalujen avulla. SQL:n avulla saatua tietoa jatkojalostetaan näillä raportointityökaluilla, jotta siitä saadaan muodostettua raportteja. Raportointiin käytettävät työkalut yleensä generoivat SQL:ää käyttäjän tarpeiden mukaan. (Hovi 2009, 19.)

### 3.4.2 SQL-kyselyt

SQL:ää käytetään tekemällä SQL-kyselyitä, joiden avulla saadaan tulosjoukko. Tietokannan hallintajärjestelmä saa kyselyitä joko vuorovaikutteisesti käyttäjän työasemalta tai suoritettava ohjelma antaa niitä. Kyselystä palautuu rivejä tietokannanhallintajärjestelmälle, joista muodostuu tulosjoukko. Lopuksi tulosjoukko palautuu työaseman käyttäjälle käyttöliittymään tai ohjelmalle käytettäväksi sen muuttujissa. (Hovi 2009, 26.)

Yksi tavallisimmista ja merkittävimmistä SQL-käskyistä on SELECT-kyselykäsky, jolla voidaan hakea tietyt sarakkeet tietystä taulusta. SELECT-kyselykäsky koostuu pakillisista SELECT-osasta, joka määrittää mitkä sarakkeet haetaan ja FROM-osasta, joka määrittää mistä taulusta haetaan. SELECT-osan määrittelemät sarakkeet tulostuvat tulosjoukkoon järjestyksessä, missä ne on lueteltu. SELECT-osassa voidaan myös luetella kaikkien FROM-osassa lueteltujen taulujen sarakkeita. Kaikki sarakkeet valitaan tähtimerkillä. Muita SELECT-kyselykäskyn osia ovat WHERE, GROUP BY, HAVING ja ORDER BY. (Hovi 2009, 27.)

SQL-kyselyt voidaan kirjoittaa isoista kirjaimista ja rivinvaihtoista välittämättä, mutta yleinen tapa on, että kyselykäskyn eri osat kirjoitetaan eri riveille ja SQL:n varaamat sanat on kirjoitettu isoilla kirjaimilla. Tämä mahdollistaa sen, että kirjoitettu koodi on huomattavasti helpommin luettavaa ja siten ymmärrettävämpää. (Hovi 2009, 25.)

Opinnäytetyössä tarvittiin paljon SQL:n käyttöä, kun tehtiin tutkimusta tärkeän tiedon sijainnista toimeksiantajan tietokannassa ja sen hyödyntämisestä. Tästä syystä kaikissa kyselyissä käytettiin SELECT-kyselyä eli saatiin tulostettua tulosjoukko haetuista tärkeistä tiedoista. SELECT-kyselyä tukena käytettiin liitoskäskyä, jotta saatiin yhdistettyä useampia tauluja samaan kyselyyn. Useamman taulun avulla oli mahdollista saada kuvaavampia ja kattavampia tulosjoukkoja. Ari Hovi (2009, 74) kirjoittaa: "Kunnolla suunnitelluissa käytännön tietokannoissa tiedot ovat kuitenkin jaakantuneet useisiin tauluihin ja tietoja on yhdisteltävä eri tauluista halutun lopputuloksen saamiseksi". Suurimmassa osassa tehdyistä kyselyistä yhdistyi kaksi tai kolme taulua.



## 4 BUSINESS INTELLIGENCE

Business Intelligence -ratkaisut mahdollistavat sen, että eri organisaatioiden ja yritysten päätöksentekijät pääsevät käsiksi liiketoimintaa tukevaan ja kuvaavaan tietoon. Tuon informaation avulla henkilöstö pystyy tekemään valistuneempia valintoja koskien liiketoimintaansa ja täten ohjaamaan sitä parempaan suuntaan. Erilaisin ratkaisuin tieto pyritään tuomaan yksinkertaisesti ja selkeästi esille, jolloin loppukäyttäjä pääsee siihen käsiksi ja pystyy käyttämään informaatiota mahdollisimman helposti. (Hovi, Hervonen, Koistinen 2009, 74.)

Erilaiset tietovarastot ovat pohjana BI-ratkaisuille, sillä niiden avulla liiketoimintatieto on hallittavassa muodossa. BI-ratkaisuista puhutaan, kun siirytään analysoimaan, esittämään ja jakelemaan informaatiota eli silloin, kun tietoa aletaan hyödyntämään sen liiketoiminnallisen ulottuvuuden näkökulmasta sekä päätöksenteon tukena. BI-ratkaisut toimivat parhaimmillaan niin, että tietojärjestelmissä piilevä, liiketoiminnan kannalta tärkeä tieto saadaan päätöksentekijöiden käytettäväksi. (Hovi ym. 2009, 74.)

Tietoa on nykyään tarjolla valtava määrä ja sitä kerätäänkin useassa yrityksessä. Ongelmana on, että tietoa ei välttämättä onnistuta hyödyntämään, vaikka esteitä ei tallennusmenetelmien ja -kapasiteetin puolesta nykypäivänä juurikaan ole. Business Intelligencen tehtävänä on vastata haasteeseen siten, että kerättyä tietoa voidaan jatkojalostaa, analysoida ja hyödyntää. (Hovi ym. 2009, 74.)

### 4.1 BI-ratkaisut yleisesti

Business Intelligence on analyyttistä, sillä sen avulla hyödynnetään erilaisista toiminnanohjausjärjestelmistä ja -ratkaisuista saatavaa informaatiota analysoimalla sitä. Business Intelligencessä käsiteltävä tieto on yleensä numeerisessa muodossa, jota tarkastellaan tiedon eri ulottuvuuksien avulla. Esimerkiksi paikan tai ajan ulottuvuuden avulla tarkasteltaessa käsiteltävä tieto saa ymmärrettävät raamit ympärilleen ja voidaan näin hyödyntää tarpeiden mukaan. (Hovi ym. 2009, 82.)

Business Intelligence on liiketoimintalähtöistä. Hovi ym. (2009, 82) kuvailevat sitä kirjassaan ”älykääksi liiketoiminnan johtamiseksi”. Business Intelligencessä on tärkeää, että sen ja liiketoiminnan yhteistyö on mahdollisimman saumatonta. Usein ongelmaksi muodostuu se, ettei liiketoiminnan ja Business Intelligencen välistä dialogia huomioida riittävästi, vaan BI-ratkaisuja tehdään teknisistä lähtökohdista. Tästä syystä on tärkeää, että BI-ratkaisuja suunnitellessa sekä IT-osasto että organisaation liiketoiminnallinen johto ovat mukana ratkaisun kehityksessä. (Hovi ym. 2009, 82.)

Vaikka BI-ratkaisujen käyttäjäryhmäksi yleensä mielletään organisaation johto, ovat ne jokaisen organisaation osan kannalta tärkeitä työkaluja. Tämä on nykyään muuttunut ja usein johto kattaa vain kolmikerroksisen käyttäjäryhmälajittelun ylimmän ja pienimmän osan. Muita tyypillisiä käyttäjäryhmiä tässä lajittelussa ovat tehokäyttäjät eli kontrollerit ja tiedon kuluttajat. Nykyään Business Intelligencen voidaan kuvailla palvelevan organisaation jokaista osa-aluetta. (Hovi ym. 2009, 82-83.)

## 4.2 BI-ratkaisujen tekniikka

Business Intelligencen tekniikka on sen historian saatossa kehittynyt ja muovaantunut kattamaan laajoja kokonaisuuksia. BI-ratkaisujen perustaksi ovat nykyään vakiintuneet tietovarastointi sekä tiedon integrointi. Tarpeen mukaan vaihtuvat analyyttiset kerrokset ovat myös osa BI-ratkaisuja, joista moniulotteinen analysointi on noussut perusraportoinnin kanssa yhtä suosituksi teknologiaksi. Tietovarastotietokanta on tukeva pohja Business Intelligence -työkaluille. Viime vuosina myös erilaiset mittaristot sekä tulokorttiratkaisut ovat alkaneet yleistymään. (Hovi ym. 2009, 85.)

BI-ratkaisujen perusarkkitehtuurilta vaaditaan nykyään kovaa kuormitusta kestävää tietovarastoa ja analysointiratkaisuja sekä tehokasta tietojen integrointia. Nämä saadaan toteutumaan erilaisin perusarkkitehtuurin komponentein, joita Hovi ym. (2009, 86.) kirjassaan luettelevat neljä. Operatiiviset tietokannat ovat yksi teknologinen komponentti perusarkkitehtuurissa, joiden tehtävänä on toimia BI-ratkaisujen raaka-aineina. Tiedon integrointi on teknologinen komponentti, jonka avulla tieto haetaan ja muokataan yhtenäiseen muotoon. Tietovarasto on paikka, johon tieto tiedon integroinnin seurauksen päätyttyä ja muodostaa näin kolmannen teknologisen komponentin. Viimeisen teknologisen komponentin muodostaa raportointi ja hyödyntäminen, ja se onkin loppukäyttäjän kannalta keskeisin komponentti. Raportoinnissa ja hyödyntämisessä määritellään se, kuinka käyttäjät saavat informaation käyttöön. (Hovi ym. 2009, 86.)

BI-ratkaisujen tärkein hyödyntämismuoto on raportointi. Nykyisessä yritysmaailmassa raportoinnin merkitys on suuri, sillä sitä käytetään ja tarvitaan kiihtyvään tahtiin. Tämän seurauksena tiedon on oltava oikeaa ja nopeammin saatavilla. BI-ratkaisujen raportointiominaisuudet ovat kehittyneet kirsityvän tahdin seurauksena tehokkaasti ja ovat nykyään usein selainpohjaisia. Perinteinen taulukkolaskentaan perustuva raportointi on kuitenkin edelleen merkityksellistä. Raportointi alkaa kyselyn eli relaatiotietokantahaun tekemisellä tietovarastoon. Jotta raportointi voitaisiin hoitaa mahdollisimman tehokkaasti, tulee kyselyjen tekemisen olla mahdollisimman helppoa loppukäyttäjän kannalta. Tärkeää olisi, että raporttien tarvitsijat voisivat itse tehdä tarvittavat kyselyt, eikä siihen vaadittaisi erillistä tiedonhallinnan tukea tai ulkopuolista konsulttia. (Hovi ym. 87.)

Yksi keskeinen ominaisuus raportointivaiheessa on käyttöliittymä, jonka kautta loppukäyttäjä tuottaa raportteja. Käyttöliittymän tulisi olla sellainen, että siinä olevat tiedot vastaavat liiketoiminnan edustajien käsityksiä tietovaraston sisällöstä. Tyypillisesti käyttöliittymä on luotu siten, että loppukäyttäjä pystyy valitsemaan, mitä tietoelementtejä eli objekteja hän haluaa kyselyynsä sisällyttää. Valittavina tietoina yleensä on niin kutsuttuja ulottuvuustietoja, joita ovat muun muassa paikka, aika ja asiakkaat sekä mittaritietoja, joita ovat esimerkiksi liikevaihto ja volyymit. Kun halutut tietoelementit on valittu, voidaan ne lisätä tulostiedoiksi eli mitä tietoja haetaan tietovarastosta. Näiden lisäksi voidaan yleensä lisätä vielä ehtoja, joilla kyseisiä tulostietoja haetaan. Kysely toteutetaan ajamalla se tietovarastosta ja sen seurauksena saadaan tulosjoukko. (Hovi ym. 87-88.)

Käyttäjän tavoitteena on luoda saadusta tulosjoukosta mahdollisimman hyödyllinen raportti. Eri keinoja raportin käyttökelpoisuuden parantamiseksi ovat muun muassa graafiset kuvaajat ja yhteenvedot. Kun raportti on muodostettu mahdollisimman käyttökelpoiseksi, sitä voidaan jakaa eri osastojen ja muiden käyttäjien välillä. Raporttiin voidaan määritellä myös erilaisia muuttuvia elementtejä, jolloin puhutaan raporttien parametrisoinnista. Parametrisoinnin avulla käyttäjä voi muuttaa raporttia antamalla sille vaihtuvat parametrit, kuten eri vuoden tai tuotteet. Parametrisoinnin avulla vakioraportointia saadaan paremmin kohdennettua. (Hovi ym. 89-90.)

#### 4.2.1 On-Line Analytical Processing

On-Line Analytical Processing eli OLAP on tietojen moniulotteista analysointia ja se on nykyään yksi BI-ratkaisujen kiinnostavimmista tekniikoista. OLAP-tekniikan avulla tieto saadaan moniulotteiseen näkymään, mikä helpottaa tiedon tarkastelua eri ulottuvuuksien kannalta ja näin sen hyödyntämistä liiketaloudelliseen analysointiin. Keskeistä tekniikalle on se, että käyttäjä pystyy suoraan hyödyntämään analysoitavaa aineistoa ilman erillisiä kyselyjä. OLAP-ratkaisuja havainnollistettaessa niiden rakennetta kuvataan usein kuution avulla. (Hovi ym. 91.)

Moniulotteisuus tarkoittaa sitä, että tietoja voidaan tarkastella eri ulottuvuuksien avulla. Esimerkiksi tietyn yrityksen myynti on voinut olla yhteensä 150 000 euroa vuodessa, mutta luvusta ei voida erottaa paljonko myynti on ollut tiettyinä eri ajanjaksoina. Yksistään tällä luvulla ei ole juurikaan minäkäänlaista analysoitavaa arvoa. Kun lisätään malliin aika uudeksi ulottuvuudeksi, saadaan myyntitulot jaettua esimerkiksi vuosineljänneksiin. Kun mallissa on eroteltuna vuosineljännesten myyntitulot, saadaan sille enemmän informatiivista arvoa. Kolmantena ulottuvuutena voisi luonnollisesti olla esimerkiksi tuote, ja näin saadaan selville, kuinka paljon kutakin tuotetta on myyty. OLAP-tekniikka mahdollistaa loputtoman määrän eri ulottuvuuksia. Yleensä pyritään kuitenkin käyttämään 3-8 ulottuvuutta, jotta kyseisen mallin hallinta ei mene liian hankalaksi. (Hovi ym. 91-92.)

Kuution avulla voidaan havainnollistaa OLAP-mallia (kuva 5). Kuution pituus, leveys ja syvyys kuvaavat eri ulottuvuuksia ja sitä, kuinka niiden avulla voidaan yksinkertaistaa tiedon hankintaa ja sen hyödyntämistä. Kuvan esimerkkiin on lisätty vielä aiemmin kuvattujen aika- ja tuoteulottuvuuksien lisäksi osastoulottuvuus, joka kertoo, kuinka paljon kukin osasto on myynyt.

KUVA 5. Kuutioesimerkki OLAP-mallista

<i>Osasto1</i>	1000	3000	3000	2500	9500
<i>Osasto2</i>	2500	4000	1500	1500	9500
<i>Osasto3</i>	1500	3000	2500	1000	8000
<b>Tuolit</b>	5 000	10 000	7 000	5 000	27 000
<b>Penkit</b>	3 000	10 000	12 000	10 000	35 000
<b>Pöydät</b>	12 000	30 000	16 000	30 000	88 000
	20 000	50 000	35 000	45 000	150 000
	<b>Q1</b>	<b>Q2</b>	<b>Q3</b>	<b>Q4</b>	<b>2017</b>

Kuution eri rivit ja sarakkeet ovat valmiiksi laskettuja. Esimerkiksi toisen vuosineljänneksen (Q2) tuolien myyntiluku saadaan, kun jokaisen osaston myyntiluvut lasketaan yhteen. Täten myös koko vuoden myyntitulo muodostuu oikeaan alanurkkaan, kun lasketaan joko kaikkien vuosineljännesten tai tuotteiden myyntiluvut yhteen. (kuva 6.)

KUVA 6. Tuolien yhteenlaskettu myyntiluku toisella vuosineljänneksellä sekä koko vuoden 2017 yhteenlaskettu myyntitulo.

	Osasto1	1000	3000	3000	2500	9500
	Osasto2	2500	4000	1500	1500	9500
	Osasto3	1500	3000	2500	1000	8000
<b>Tuolit</b>		5 000	10 000	7 000	5 000	27 000
<b>Penkit</b>		3 000	10 000	12 000	10 000	35 000
<b>Pöydät</b>		12 000	30 000	16 000	30 000	88 000
		20 000	50 000	35 000	45 000	150 000
		<b>Q1</b>	<b>Q2</b>	<b>Q3</b>	<b>Q4</b>	<b>2017</b>

#### 4.2.2 FASMI-määritelmä

OLAP-ratkaisuista on kehitetty erilaisia määritelmiä. Ne määrittelevät, kuinka OLAP-ratkaisuja käytetään, mitkä ovat niiden keskeisiä ominaisuuksia ja mitä niiden on tarkoitus tarjota käyttäjälle. Yksittäinen määritelmä on FASMI eli Fast Analysis of Shared Multidimensional Information. Nimi tarkoittaa nopeaa jaetun tiedon analysointia moniulotteisesti. (Hovi ym. 2009, 93.)

OLAP-ratkaisujen ehkä tärkein yksittäinen tekijä on nopeus ja siksi se kuuluu FASMI-määritelmään. Määritelmän mukaan vastaus pitää saada alle viidessä sekunnissa FASMI-testissä. Kuten kaikissa BI-ratkaisuissa, tiedon analysointi on tärkeää ja näin FASMI:ssa on määritelty, että OLAP-ratkaisun täytyy sisältää riittävästi liiketoimintalogiikkaa. Tiedon jakaminen on keskeistä analysoinnin ja sen hyödyntämisen kannalta, joten on myös määritelty, että jakamisen tulee olla osa OLAP-ratkaisua ja siinä tulee olla huomioituna toimiva tietoturva.

#### 4.2.3 Tietovarastot

Operatiiviset järjestelmät eivät sovellu parhaiten tiedon käyttämisessä raportointiin ja analysointiin, joten tarvitaan kyseiseen käyttöön erityisesti suunniteltu erillinen tietokanta, tietovarasto. Tietovarastotietokantaan ei päivitetä tietoja suoraan, vaan tieto saadaan perusjärjestelmien käyttämistä tietokannoista. Perusjärjestelmien tietokantojen raakatieta jalostetaan raportoitavaan ja kyselykäyttöön soveltuvaan muotoon. (Hovi ym. 2009, 14.)

Raakatieta siirretään perusjärjestelmien tietokannoista tietovarastoon ETL-prosessin avulla. ETL-prosessissa raakatieta jalostetaan käytettävämpään muotoon, jotta tietovarastossa oleva tieto voidaan käyttää hyödyksi tehokkaasti. Prosessissa tietoa yhdenmukaistetaan ja yhdistetään, jotta se saadaan tietovaraston kannalta tarpeelliseen ja oikeanlaiseen muotoon. Esimerkkinä sukupuolikoodit, jotka voivat olla perusjärjestelmien tietokannoissa muun muassa "M" ja "N" tai "1" ja "0", voi-

daan yhdenmukaistaa niin, että käytetään tietovarastossa vain "M" ja "N", jolloin kaikki muut vaihtoehdot muunnetaan ETL-prosessi vastaamaan sitä. Kun prosessi on valmis, tieto siirretään tietovarastoon. (Hovi ym. 2009, 14 – 15.)

Tietovarastolla on merkittäviä etuja tavanomaiseen tietokantaan nähden. Tietovarastoissa tieto on integroitua eli sen sisältämä tieto on käytettävissä yhdestä paikasta. Tietovaraston on riippumaton liiketoiminnan prosesseista, mikä mahdollistaa sen, ettei niitä tarvitse muuttaa tietovarastoa suunniteltaessa ja toteutettaessa. Tämän seurauksena myös raportointi pystyy olemaan joustavampaa ja tehokkaampaa. Tietovarastoissa tiedot on tallennettu helposti ja nopeasti kyseltäväksi, minkä lisäksi niihin on tallennettuna myös metatieto eli kuvaavat tiedot. (Hovi ym. 2009, 15.)

#### 4.3 Business Intelligencen merkitys

Erilaisia järjestelmiä päätöksenteon tueksi on kehitetty käytännössä siitä asti, kun tietokoneet tulivat mukaan yritysten toimintaan. Vaikka järjestelmät ovat vuosien varrella muuttaneet nimeään, on niiden perustavoite ja -idea säilynyt nykypäivään. Varsinaisen Business Intelligencen historian katsotaan alkaneen 1989, kun Howard Dresner käytti kyseistä termiä edellä mainittua teknologiaa kuvaessaan. Nykyään Business Intelligencen merkitys erilaisissa organisaatioissa korostuu koko ajan. Päätöksiä on pystyttävä tekemään nopeammin, digitaalisen tiedon määrä kasvaa kovaa tahtia ja viranomaisvaatimukset ovat lisänneet tehokkaan raportoinnin tarvetta. Nämä kaikki ovat haasteita, joihin Business Intelligence on yksi keino tuoda ratkaisuja. (Hovi ym. 2009, 77.)

Business Intelligencen tärkein tavoite on toimia päätöksenteon tukena. Hovi ym. (2009, 82) kirjoittavat aiheesta kirjassaan: "Lähtökohtaisesti on edelleen kysymys paremman informaation saattamisesta käyttäjille ja oikeampien päätösten tekemisestä epätäydellisen informaation vallitessa". Ajatus päätöksenteon tukemisesta pohjautuu jo Business Intelligencen edeltäjiin, sillä niitä kutsuttiin päätöksenteon tukijärjestelmiksi. (Hovi ym. 2009, 80.)

Tiedon tulee olla oikeellista ja ajankohtaista sekä käyttäjän tietotarpeita vastaavaa, jotta tärkein tavoite saadaan toteutumaan. Tilanteet, joissa analysoitavaa tietoa tarvitaan, voivat olla ennalta määriteltyjä tai ennalta määrittelemättömiä. Ennalta määrittelemättömien tilanteiden pohjalta voidaan tarkastella BI-ratkaisujen varsinaista käyttökelpoisuutta. BI-työkalut, jotka mahdollistavat kätevät kysely- ja raportointiominaisuudet, auttavat ennalta määrittämättömiin tilanteisiin vastaamisessa. Näiden ominaisuuksien pohjalta useat BI-ratkaisut ovat kehittyneet, minkä seurauksena ne vastaavat hyvin tavoitteeseen tietotarpeiden täyttämisestä. (Hovi ym. 2009, 80.)

Yrityksissä on perinteisesti raportoitu mennyttä aikaa ja sen pohjalta tehty ratkaisuja. Business Intelligence yhdistettynä suorituskyvyn hallintaan on kuitenkin muuttamassa tätä niin, että huomio kiinnittyy enemmän nykyhetken seuraamiseen ja tulevan ennakkointiin. Yksi Business Intelligencen tavoitteista on tukea organisaation omia tavoitteita ja strategiaa mahdollisimman tehokkaasti. Nykyhetken ja tulevaisuuden valjastaminen hyötykäyttöön Business Intelligencen avulla toteutetaan eri-

laisin ratkaisuin, kuten tunnuslukuja seuraavilla mittaristoratkaisuilla ja monipuolisilla suunnittelutyökaluilla. Paras hyöty Business Intelligencestä saadaan, kun se on yhdistettynä kvantitatiivisiin eli määrällisiin mittareihin. (Hovi ym. 2009, 80-81.)

Business Intelligence yksinkertaistaa tiedonhyödyntämisprosessia. Yleisenä ongelmana kerätyn tiedon hyödyntämisessä on ollut se, ettei näitä tietoja tarvitseva käyttäjä ole osannut tai kyennyt tekemään tietotarpeita vastaavia hakuja tietokantoihin. On tarvittu erillinen IT-osasto organisaatioon, joka toimii välikätenä tiedon hyödyntämisessä. Tämä on luonut viivettä tiedonhyödyntämisprosessiin ja siten heikentänyt tiedon hyödyntämisen tehokkuutta. Business Intelligencen tavoitteena on yksinkertaistaa tuota prosessia ja mahdollistaa käyttäjän omatoiminen tiedonkeruu silloin, kun he sitä tarvitsevat. (Hovi ym. 2009, 81.)

Myynnin lisääminen, kustannuksissa säästäminen ja uusien liiketoimintamahdollisuuksien avaaminen ovat liike-elämän investointien tavoitteita, jotka voidaan luetella myös BI-ratkaisujen tavoitteiksi. BI-ratkaisujen tehokkuutta voidaan mitata sillä, miten nopeasti ja tehokkaasti oikeanlainen tieto tavoittaa sitä tarvitsevat käyttäjät. Kun tuossa välissä syntyvää viivettä saadaan pienemmäksi, lisätään toiminnan operatiivista tehokkuutta ja säästetään kustannuksissa. BI-ratkaisujen vaikeammin mitattavia etuja ovat paremman päätöksenteon kautta syntyneet hyödyt ja niistä seuranneet oikeammat ratkaisut yrityksen jokapäiväisessä toiminnassa (Hovi ym. 2009, 81.)

## 5 TOIMINNANOHJAUSJÄRJESTELMÄ

Toiminnanohjausjärjestelmä on laaja yrityksen ohjaamiseen tarkoitettu kokonaisvaltainen tietojärjestelmä. Toiminnanohjausjärjestelmiä kutsutaan myös ERP-järjestelmiksi. Nimitys tulee sanoista "Enterprise Resource Planning" ja tarkoittaa sananmukaisesti yrityksen resurssien suunnittelua. Järjestelmälle on tyypillistä, että sen eri toiminnot ovat integroituja eli ne käyttävät samaa tietokantaa. Integraatio mahdollistaa ajantasaisen tiedon saatavuuden kaikille järjestelmän toiminnoille, mikä tehostaa yrityksen toimintaa ja taloudellisuutta. (Logistiikan maailma.)

### 5.1 Toiminnanohjausjärjestelmien merkitys ja keskeiset ominaisuudet

Toiminnanohjausjärjestelmät ovat kehittyneet materiaalien tarvelaskennan pohjalta. Tämän pohjalta on kehitetty laajempia järjestelmäkokonaisuuksia, jotka kattavat taloushallinnon ja tuotannonohjaus-toiminnot. Toiminnanohjausjärjestelmät ovat nykyään kasvaneet kattamaan lähes kaikki yrityksen keskeiset toiminnot, kuten kirjanpidon, varastonhallinnan, tuotannon ja henkilöstöhallinnon. (Logistiikan maailma.)

Toiminnanohjausjärjestelmien tavoitteena on parantaa yrityksen toiminnan tehokkuutta, taloudellisuutta, läpinäkyvyyttä ja asiakaspalvelua. Toiminnanohjausjärjestelmien avulla yritys pystyy karsimaan päällekkäistä työtä sekä parantamaan käyttöastetta ja näin ollen tehostamaan yrityksen toimintaa. Taloudellisia hyötyjä toiminnanohjausjärjestelmällä saavutetaan, kun mahdollistetaan tehokkaampi materiaalihojaus sekä alhaisempi varastonkäyttö. Sama tieto on käytössä yrityksen eri toiminnoilla, mikä lisää läpinäkyvyyttä. Myös asiakaspalvelu paranee, kun lupaukset toimitusajoista perustuvat reaaliaikaiseen tietoon. (Logistiikan maailma.)

Yksi toiminnanohjausjärjestelmän keskeisistä ominaisuuksista on sen reaaliaikaisuus. Reaaliaikaisuus mahdollistaa yrityksen eri toimintojen yhtäaikaisen reagoinnin muuttuviin tilanteisiin ja näin ollen tehostaa toimintaa. Reaaliaikaisuus asettaa myös haasteensa tiedon oikeellisuudelle, sillä jatkuvasti muuttuvan ja syntyvän perustiedon on oltava ajantasaista sekä oikeaa. Esimerkiksi tuotettujen tuotteiden määrä on kirjattava ajoissa ja oikein. (Logistiikan maailma.)

## 6 TUTKIMUS JA TOTEUTUS

Opinnäytetyön keskeisenä tutkimuskysymyksenä täytyi selvittää, mikä tieto toiminnanohjausjärjestelmässä ja sen käytössä on asiakkaiden kannalta tärkeää ja miten sitä voidaan hyödyntää yhteenvetosovelluksessa. Opinnäytetyön tutkimusosan tavoitteena oli tehdä selvitys näistä tiedoista ja siitä, missä ne sijaitseva tietokannassa ja miten ne ovat hyödynnettävissä. Tutkimuksen tarkoituksena on toimia pohjana kehitettävälle yhteenvetosovellukselle.

Opinnäytetyön toteutusosan tavoitteena oli kehittää SQL-kyselyt, joilla saatiin aiemmin tutkimuksessa selvitettyjä tietoja tietokannasta. Lisäksi toteutusosan tavoitteena oli myös kehittää itse yhteenvetosovellusta, jonka avulla tuotetaan toimeksiantajan asiakasyritysten johdolle yhteenvetoja toiminnanohjausjärjestelmän käytöstä. Tarkoituksena oli saada sovellus kehitettyä niin pitkälle, kuin opinnäytetyön aikana ehdittäisiin.

Tutkimus alkoi perehtymällä yleisesti toimeksiantajan työympäristöön ja sen tarjoamaan palveluun eli toiminnanohjausjärjestelmään. Perusteisiin perehtymisen jälkeen tehtiin listaus tiedoista, joita tarvitaan yhteenvetojen tekemiseen. Tämän jälkeen tutkittiin toiminnanohjausjärjestelmän taustalla toimivaa tietokantaa sekä tutustuttiin tietokantojen teoriaan, jonka avulla pystyin syventämään opinnoissa oppimaani tietoa. Tietokantoihin tutustumisen jälkeen selvitettiin, missä aiemmin listatut tärkeät tiedot tietokannassa sijaitsevat ja tehtiin SQL-kyselyitä tietokantaan, joiden avulla tärkeät tiedot saatiin hakutuloksiksi.

### 6.1 Perehdytys ja tutkimus

Tutkimustyön ensimmäisenä vaiheena oli perehdytys toimeksiantajan tarjoamaan palveluun eli toiminnanohjausjärjestelmään. Käyttäjätunnusten avulla pystyttiin tutustumaan tuotteen ominaisuuksiin. Tutkimuksessa pyrittiin ymmärtämään asiakkaan näkökulmaa järjestelmässä, jotta voitaisiin hahmottaa, mikä tieto järjestelmän käytössä on tärkeää. Pohdittiin, mitä asiakkaat tarvitsevat järjestelmästä, miten he näkevät järjestelmän eri toiminnallisuudet sekä niiden käytettävyyden ja kuinka he pystyvät hyödyntämään saatavaa liiketoimintatietoa yleisesti.

Toiminnanohjausjärjestelmän lisäksi tutkimuksessa täytyi perehtyä myös työympäristöön ja käytettäviiin työkaluihin. Opinnäytetyön tutkimusosan alussa perehdyttiin Linux-ympäristöön, sillä ensimmäisessä opinnäytetyön tekoon käytetyssä päätelaitteessa oli Linux-pohjainen Antergos käyttöjärjestelmä. Vaikka myöhemmin päätelaite vaihtui sellaiseen, jossa käyttöjärjestelmänä oli Windows, tarvittiin silti Linux-terminaalin osaamista. Tietokannan käyttöön saatiin tarvittavat avaimet ja aluksi käytettiin MySQL Workbench -työkalua tietokantojen käsittelyyn. Myöhemmin päätelaitteen vaihtuessa siirryttiin käyttämään HeidiSQL:ää visuaalisena tietokantojen käsittelytyökaluna.

Opinnäytetyön tarkkaa laajuutta ei etukäteen määritelty, mutta sovittiin, että toteutuksen rajaus muotoutuu työn edetessä. Opinnäytetyö alkoi perehtymällä työympäristöön ja järjestelmään sekä



selvittämällä, mikä tieto toiminnanohjausjärjestelmässä on asiakkaan kannalta tärkeää, ja jota voitaisiin hyödyntää yhteenvetosovelluksessa. Lisäksi keskusteltiin useiden työntekijöiden kanssa tarvittavista tiedoista ja näin saatiin kartoitettua, mitä tietoa tietokannoista lähdetään tutkimaan.

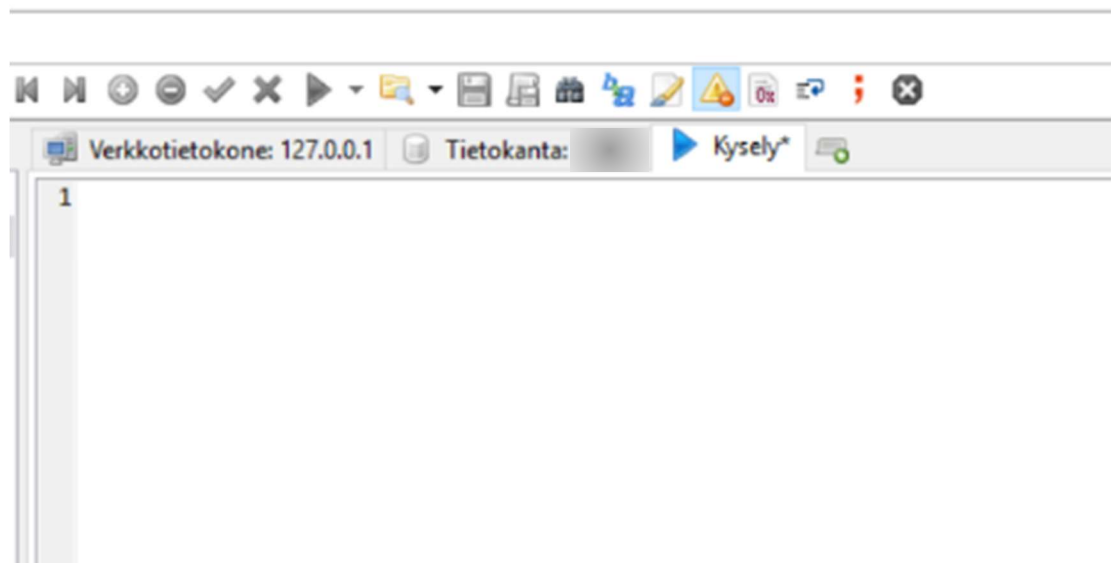
Yhteenvetosovelluksen kannalta keskeiset tiedot jaettiin kolmeen pääkategoriaan: työaika, tehtävät ja telematiikka. Nämä kolme kategoriaa liittyvät toiminnanohjausjärjestelmän keskeisiin ominaisuuksiin, joten pääkategorioiden rajaaminen tällä tavoin oli luonnollista. Pääkategoriat jaettiin määrittelemällä tarkemmin, mitä dataa kerätään kustakin pääkategoriasta. Kaikki kerättävät tiedot listattiin kategorioittain, jotta toteutusvaiheessa olisi helpompaa seurata, mitkä tiedot on jo haettu ja mitä tulee vielä hakea.

Työajasta kokonaistunnit, keskiarvoinen kesto ja ajoneuvojen työaika olivat keskeisiä tutkittavia tietoja yhteenvetosovellusta varten. Kokonaistunnit määriteltiin kerättäviksi käyttäjäryhmittäin, kuukausittain, kustannuspaikoittain ja palkkalajeittain. Tehtävistä keskeisimmäksi tiedoksi määriteltiin asiakaskohtaiset tehtävämäärät. Muita tehtäviin liittyviä kerättäviä tietoja ovat niissä toimitetut tuotteet, ajatut kilometrit, tehdyt tunnit ja yrityksen tehtävämäärät tietyn ajanjakson aikana. Telematiikan osalta ajoneuvojen kilometrit, tunnit, käyttöaste, ajotilat, katsastukset sekä tankkaukset listattiin kerättäviksi tiedoiksi.

## 6.2 Tietokanta

Opinnäytetyön toteutuksen ensimmäisenä vaiheena oli tutustua toimeksiantajan järjestelmän taustalla olevaan tietokantaan. Tietokanta on yrityksen toiminnan aikana kasvanut suureksi, joten tutkiminen alkoi tutkimalla visuaalisessa käyttöliittymässä eri taulujen tietoja, niiden muotoja sekä yhteyksiä eri taulujen välillä. Tietokannan visuaaliset käsittelytyökalut MySQL Workbench sekä HeidiSQL mahdollistavat taulun tutkimisen niin, että käyttäjä pystyy näkemään esimerkiksi taulun avaimet, sarakkeiden tietotyypit ja yhteydet muiden taulujen avaimiin. Molemmilla työkaluilla pystyy myös tutkimaan tauluja rivitasolla ohjelman generoidessa automaattisesti haun, kun taulu valitaan. (Kuva 7.)

KUVA 7. HeidiSQL:n SQL-editorin käyttöliittymä



Tietokantaan perehdyttiin tutkimalla tietokannan eri taulujen dataa ja pyrittiin jäsentämään, millä tavoin ne kytkeytyvät toiminnanohjausjärjestelmään. Kun oli saatu riittävän hyvä yleiskuvan tietokannassa olevista tiedoista, siirryttiin tekemään erilaisia SQL-kyselyitä. SQL-kyselyiden avulla pystyttiin syventämään ymmärrystä toimeksiantajan tietokannasta ja siellä olevasta tiedosta.

Kun ymmärrys tietokannasta alkoi olla riittävällä tasolla, siirryttiin tekemään kyselyitä niin, että ne palauttivat aiemmin määriteltäviä keskeisiä tietoja toiminnanohjausjärjestelmän käytöstä. Tarvittavat tiedot oli listattu perehtymisvaiheessa ja tämän listan mukaan tehtiin kyselyitä, jotka palauttivat listassa määritellyt halutut tiedot. Kun kyselyt olivat siinä muodossa, että ne palauttivat haluttuja tietoja, tallennettiin ne erilliseen listaan, jotta pystyttiin seuraamaan, että kaikki halutut tiedot saadaan SQL-kyselyillä tietokannasta.

Kun lista alkoi olla valmis eli jokainen haettava tieto oli tietokannasta eri SQL-kyselyillä saatu palautettua, siirryttiin kyselemään asiantuntijoilta tarkennuksia kyselyihin. Tarkennusta tarvittiin erityisesti siihen, että tiedot olivat varmasti oikeista tauluista, ja että oli ymmärretty haettava tieto sekä sen palautuminen oikein. Asiantuntijoiden tarkastaessa kyselyitä selvisi useita seikkoja, joita pystyttiin kyselyissä parantamaan ja korjaamaan. Korjatut SQL-lauseet toimivat pohjana opinnäytetyön seuraavalle vaiheelle, kun tietokannasta haettuja tietoja alettiin hyödyntämään.

### 6.3 Sovelluskehitys

Kun tietokannasta oli erilaisten kyselyiden avulla saatu palautettua halutut tiedot, siirryttiin kehittämään yhteenvetosovellusta eli loppusovellusta. Toteutuksen sovelluskehitysvaiheessa ensimmäisenä perehdyttiin toimeksiantajan ohjelmointiympäristöön ja siinä käytettäviin tekniikoihin. Aluksi täytyi luoda käyttäjä sekä muodostaa yhteys kehityspalvelimelle. Kun yhteys oli muodostettu, kehityspalvelimesta käytiin läpi perusteet eli se, miten sinne voidaan lisätä uutta koodia. Lisäksi perehdyttiin

myös siihen, miten kehityspalvelimella sijaitsevaa koodia voidaan testata ja näin selvittää mahdolliset koodissa esiintyvät virheet.

Seuraavana vaiheena alettiin kehittämään PHP-moduuleja. Jokaiselle SQL-kyselylle kehitettiin oma vastaava PHP-moduuli eli esimerkiksi tietyn käyttäjän tehtävien määrän palauttava kysely on tietyssä PHP-moduulissa. Moduulien on tarkoitus toimia välikappaleina loppusovelluksen ja tietokantaan tehtävien SQL-kyselyiden välillä. Tämä ratkaisu tuo sovellukselle joustavuutta, kun itse loppusovelluksessa ei suoraan toteuteta SQL-kyselyitä. Osana PHP-moduulien kehittämistyötä oli myös selvittää, onko joistakin kerättävistä tiedoista jo olemassa moduuleja, jotka palauttavat vastaavaa dataa.

PHP-moduulit ovat käytännössä funktioita, joten loppusovellus pystyy kutsumaan PHP-moduulia palauttamaan dataa tarpeen mukaan. PHP-moduulit koostuvat tyypillisestä PHP-funktiosta, joka toteuttaa ohjelmointikieleen upotetun SQL-kyselyn annetuilla parametreilla, ottaa vastaan kyselystä saadun datan ja palauttaa sen oliona JSON-formaatissa (kuva 8). PHP-moduulien kehittämiseen käytettiin Notepad++ -lähdekoodieditoria.

KUVA 8. Esimerkki PHP-moduulista Notepad++ -lähdekoodieditorissa.

```
public function getExample() {
    $testId = $this->r()->getResourceParam( 'exampleId' );

    $query = $this->getEntityManager()->getConnection()->prepare("SELECT * FROM testTable tt
WHERE tt.exampleValue = :parameter");
    $query->execute( array(
        'parameter' => $testId
    ));

    $result = $query->fetchAll();

    return $this->jsonResponse( $result );
}
```

### 6.3.1 Dokumentointi

PHP-moduulit kehitettiin Kihon rajapintaan, joten ne täytyi myös dokumentoida asianmukaisesti. Kihon rajapinta on nimeltään KihoAPI ja se pohjautuu OpenAPI -ohjelmointirajapintaan. Varsinainen dokumentointi tapahtui kommentoimalla koodia määritellyn formaatin mukaan. Kommentoinnin avulla pystyttiin määrittelemään kyseisen funktion nimi ja kuvaus sekä funktion palauttamien olion ominaisuuksien kuvaukset ja tietotyypit dokumentointia varten (kuva 9).

KUVA 9. Esimerkki dokumentoinnista kommentoimalla koodia.

```

/**
 * @API\Request (
 *     method="GET",
 *     summary="Get example data",
 *     description="Returns example data from test server",
 *     parameters={
 *         @API\Parameter(
 *             in="PATH",
 *             schema=@API\Schema(
 *                 ref="\TestObject\Objects\Object::id"
 *             )
 *         ),
 *     },
 * )

```

Kihon rajapinnassa dokumentointi tapahtuu määritellyn formaatin avulla kommentoimalla koodia. Dokumentoitavalle koodille määritellään nimi ja kuvaus, sekä mahdolliset lisätiedot, kuten ominaisuuksien tietotyypit. Kihon rajapinnan visuaalista käyttöliittymää voidaan tarkastella selaimessa, jossa koodin dokumentointi on tarkasteltavissa. Dokumentoinnin avulla rajapinnan tarkastelija pystyy näkemään kyseisen toiminnallisuuden tiedot ja ominaisuudet.

### 6.3.2 Testaus

Jotta PHP-moduulien toimivuus voitiin varmistaa, tuli niitä testata. PHP-moduulien kehitys alkoi rajapinnan ulkopuolella, jolloin testaaminen tapahtui yksinkertaisesti palauttamalla dataa selaimessa. Kun PHP-moduulia eli funktiota kutsuttiin lataamalla se selaimen, palautti se JSON -formaatisia dataa taulukkomuotoisena. Datan oikeellisuus varmistettiin siten, että selaimen palautunutta dataa verrattiin SQL-kyselyn palauttamaan dataan ja mikäli tiedot olivat yhtenevät, moduuli toimi oikein.

Moduulien toimivuus myös Kihon rajapinnassa sekä dokumentoinnin oikeellisuus oli testattava. Testaus tapahtui Kihon rajapinnassa ja siinä lähinnä tarkistettiin, saatiinko dokumentointi ja moduulit näkymään oikein rajapinnan käyttöliittymässä. Ongelmia ilmeni ensimmäisillä testauskerroilla, kun käyttöliittymään ei onnistuttu saamaan näkyviin useampaa moduulia kerrallaan. Ongelma liittyi kuitenkin rajapinnan taustalla olevaan koodiin ja se korjattiin. Testauksen lopputuloksena saatiin Kihon rajapinnan käyttöliittymään asianmukainen dokumentointi.

## 7 YHTEENVETO

### 7.1 Jatkokehitys

Opinnäytetyön edetessä päätettiin, että toteutus rajataan PHP-moduulien kehittämiseen, joten yhteenvetosovelluksen lopullisen käyttöliittymän ja toiminnallisuuden kehitys siirtyy opinnäytetyön ulkopuolelle. Kehityksen seuraavana vaiheena on suunnitella yhteenvetosovelluksen toiminnallisuudet ja tekniikka. Tarkoitus on, että yhteenvetosovellus pystyy hyödyntämään opinnäytetyön toteutuksessa kehitettyjä PHP-moduuleja mahdollisimman saumattomasti. Yhteenvetosovelluksessa käytettävät tekniikat tulevat olemaan oletettavasti pitkälti samankaltaisia, mitä opinnäytetyön toteutuksen sovelluskehitysvaiheessa käytettiin.

Kokonaisuudessaan yhteenvetosovelluksen sijasta voidaan puhua myös yhteenvetojärjestelmästä. Toimeksiantajan tavoitteena on luoda järjestelmä, joka lähettää asiakasyritysten johdolle automaattisesti luodun sähköposti, jossa on tiivistettynä keskeisimmät yhteenvedot. Sähköpostissa on tarkoitus olla linkki varsinaiseen yhteenvetosovellukseen, jossa käyttäjä pääsee tarkastelemaan erilaisia yhteenvetoja syvällisemmin.

### 7.2 Henkilökohtaisen osaamisen kasvu

Opinnäytetyön aikana jouduin hyödyntämään ja kehittämään omaa henkilökohtaista osaamistani monilla eri tavoin. Projektityöskentelytaidot olivat keskeisiä, kun suunnittelimme opinnäytetyön toteutusta ja selvitimme, mikä tieto toiminnanohjausjärjestelmän käytössä on asiakkaan kannalta keskeistä. Opinnäytetyön aikana tein yhteistyötä usean eri henkilön kanssa ja tämä myös vaati kykyä toimia osana projektiryhmää. Opinnäytetyön toteutuksessa tarvitsin opiskeluissani oppimia ohjelmointitaitoja erityisesti tietokantojen ja PHP:n osalta. Opinnäytetyön raportointiosiossa vaadittiin myös kykyä tuottaa kielellisesti ehjää sekä sujuvaa tekstiä.

Projektityyppistä opinnäytetyötä tehdessä myös aikataulutusta oli keskeisessä roolissa. Aikataulutuksen merkitys korostui entisestään, kun aloin tekemään toimeksiantajalle osa-aikaista työtä opinnäytetyön lisäksi. Oli tärkeää pystyä hoitamaan osa-aikaisen työn työtehtävät riittävän tehokkaasti sekä edistää opinnäytetyötä samanaikaisesti, joten näiden kahden keskinäinen aikatauluttaminen nousi tärkeään rooliin. Kahden eri työn hoitaminen vaati myös kykyä vaihtaa ripeästi työstä toiseen. Sekä opinnäytetyöhön että osa-aikaiseen työhön liittyi paljon oppimista molempiin, joten kyky ottaa vastaan tietoa ja kehittää omaa osaamistaan oli tärkeässä roolissa.

Edellä mainitut henkilökohtaiset taidot kehittyivät opinnäytetyön aikana huomattavasti. Alun epävarmuus muuttui etenemisen ja onnistumisten myötä tehokkaaksi työskentelyksi. Opinnäytetyön aikana opin paljon siitä, kuinka selvittää ja edetä stressaavista työtilanteista. Ammatillinen osaaminen ja itsevarmuus kehittyivät kevään aikana opinnäytetyötä tehdessä, mikä on valtava apu työelämään siirryttäessä.

### 7.3 Tavoitteiden saavuttaminen

Vaikka opinnäytetyön edetessä toteutuksen lopullinen laajuus rajattiin alkuperäistä suunnitelmaa kapeammaksi, oli se silti riittävän kattava opinnäytetyöksi ja ehjä pohja suunniteltua yhteenvetosovellusta varten. Kevään aikana sain toimeksiantajan yrityksestä opinnäytetyön lisäksi osa-aikaisen työn, minkä seurauksena työaika jakaantui näiden kahden kesken. Kahden työn aikataulutukset vaikuttivat opinnäytetyön etenemiseen ja tämän pohjalta tehtiin päätös opinnäytetyön toteutuksen rajaamisesta. Opinnäytetyössä onnistuttiin selvittämään, mikä tieto toiminnanohjausjärjestelmän käytössä on asiakasyritysten kannalta tärkeää sekä tutkimaan, miten kyseinen tieto on hyödynnettävissä tietokannasta. Lisäksi onnistuttiin kehittämään PHP-moduulit, jotka toimivat pohjana myöhemmin kehitettävälle yhteenvetosovellukselle.

## LÄHTEET JA TUOTETUT AINEISTOT

- LOGISTIIKAN MAAILMA. Toiminnanohjausjärjestelmä [Verkkoaineisto]. [Viitattu 2018-1-18] Saatavissa: <http://www.logistiikanmaailma.fi/logistiikka/ohjausjarjestelmat/toiminnanohjausjarjestelma/>
- KAUPPAPUISTO. Liiketoimintatiedot hallinta (BI) [Verkkoaineisto]. [Viitattu 2018-1-18.] Saatavissa: <https://www.kauppapuisto.net/BI.php>
- ITEWIKI. BI (Business Intelligence) ja raportointi [Verkkoaineisto]. [Viitattu 2018-1-18.] Saatavissa: <https://www.itewiki.fi/opas/bi-business-intelligence-ja-raportointi/>
- KIHO 2018a. Yritys [Verkkojulkaisu]. [Viitattu 2018-1-31.] Saatavissa: <https://www.kiho.fi/yritys/>
- KIHO 2018b. Kiho-tuotteet [Verkkojulkaisu]. [Viitattu 2018-1-31.] Saatavissa: <https://www.kiho.fi/category/tuotteet/>
- KIHO 2018c. Tuotteiden ominaisuudet [Verkkojulkaisu]. [Viitattu 2018-1-31.] Saatavissa: <https://www.kiho.fi/tuotteiden-ominaisuudet/>
- HOVI, Ari, HUOTARI, Jouni, LAHDENMÄKI, Tapio 2005. Tietokantojen suunnittelu & indeksointi. 1. painos. Porvoo: WS Bookwell.
- EKONOJA, Antti, LAHTONEN, Tommi, MÄNTYLÄ, Jukka, 2004. Relaatiokantojen peruskäsitteet [Verkkoaineisto]. [Viitattu 2018-2-12.] Saatavissa: <http://appro.mit.jyu.fi/doc/tiedonhallinta/tietokannat/index2.html>
- HOVI, Ari, HERVONEN, Henriikki, KOISTINEN, Heikki, 2009. Tietovarastot ja Business Intelligence. 1. painos. Porvoo: WS Bookwell.
- HOVI, Ari 2009. SQL-Opas. 8. painos. Jyväskylä: Docendo Finland Oy
- MICROSOFT 2017. Tietokannan normalisoinnin perusteiden kuvaus [Verkkoaineisto]. [Viitattu 2018-3-16] Saatavissa: <https://support.microsoft.com/fi-fi/help/283878/description-of-the-database-normalization-basics>
- MARIADB FOUNDATION 2018. About MariaDB [Verkkoaineisto]. [Viitattu 2018-4-16.] Saatavissa: <https://mariadb.org/about/>